Structured Output Learning with Candidate Labels for Local Parts*

Chengtao Li[1], Jianwen Zhang**[2], and Zheng Chen[2]

[1] Institute for Interdisciplinary Information Sciences,
Tsinghua University, Beijing, China, 100080
lichengtao2010@gmail.com,
[2] Microsoft Research Asia, Beijing, China, 100080
jiazhan@microsoft.com, zhengc@microsoft.com

Abstract. This paper introduces a special setting of weakly supervised structured output learning, where the training data is a set of structured instances and supervision involves candidate labels for some local parts of the structure. We show that the learning problem with this weak supervision setting can be efficiently handled and then propose a large margin formulation. To solve the non-convex optimization problem, we propose a proper approximation of the objective to utilize the Constraint Concave Convex Procedure (CCCP). To accelerate each iteration of CCCP, a 2-slack cutting plane algorithm is proposed. Experiments on some sequence labeling tasks show the effectiveness of the proposed method.

Keywords: Structured Output Learning, Weak Supervision, Candidate Labels, Local Parts

1 Introduction

Many applications involve predicting structured labels for a set of interdependent variables. For example, a part-of-speech tagging (POS) model needs to predict a sequence of POS tags for a sentence, one for each token. This type of problem is known as *structured output learning*. In the past decade, some effective methods have been proposed and widely used, such as Conditional Random Field (CRF) [21] and Structured SVM (SVM^{struct}) [33]. However, they are supervised methods requiring a large amount of labeled instances for training, which are expensive due to the natural complexity of the output, e.g., each token of a sentence needs labeling. Although some semi-supervised [44] and active learning methods [26, 27] are proposed to reduce the number of required labels, they still require *exact* labels for the output variables. In reality, while getting exact labels as supervision is expensive, it is often cheap to get much weak/indirect supervision, e.g., some candidate labels for an instance. Thus utilizing weak/indirect

* Please see http://research.microsoft.com/pubs/194791/SupplementaryMaterial.pdf for all the proofs and more details of the algorithms.

** The contact author.

supervision to train a high quality predictor is very meaningful [3, 14, 18, 35]. In this paper, we introduce a special setting, called *Candidate Labels for Local Parts* (CLLP). In CLLP, for each instance, we only need to provide a set of candidate labels for each local part of the output variables (e.g., a chunk in a sequence), among which only one is correct.

The CLLP setting takes root in many real-world scenarios, which roughly falls into two cases: (1) There is prior knowledge from which we can provide a candidate labeling set for a local part of output variables. For example, for POS tagging, by looking up some linguistic dictionaries, we can get the candidate POS tags for a word in a sentence [23, 28, 25, 9]. Similar scenarios exist for other sequence labeling tasks like word sense disambiguation [24], entity recognition [31], etc.. Another example is caption based image auto tagging. An image on the web is usually surrounded with tags that provide candidate labels for objects in the image [1, 4, 13]. (2) Noisy labels from multiple annotators. When we collect manual labels for a learning task, a labeling task is often assigned to multiple annotators, e.g., via a Crowdsourcing system. Due to labeling bias or irresponsible annotators, different annotators may give different labels for the same output variable. Thus the annotators collectively provide candidate labels for an output variable [7].

CLLP also provides a uniform viewpoint for different labeling settings of structured output learning: (1) If the candidate labeling set for each output variable is the full label space, i.e., all the possible labels, there is no useful information provided and hence it degenerates to unsupervised learning. (2) If the candidate labeling set for each output variable contains only the ground truth label, it degenerates to fully supervised learning. (3) If for some instances, we provide the candidate labels as (1) and for other instances we provide the candidate labels as (2), then it becomes semi-supervised/transductive learning [44]. (4) The general case is that each local part of the output variables is assigned with a non-trivial set of candidate labels.

In this paper, we propose a large margin approach to the CLLP setting. We maximize the margins between candidate labels and non-candidate labels, and also the margins between the predicted label and other candidate labels. Since the obtained optimization problem is non-convex, the proper approximations and Constraint Concave Convex Procedure (CCCP) are used to solve it.

The major contributions of this paper are as follows:

1. We introduce and formalize CLLP, a general type of weakly supervised setting for structured output learning and propose a large-margin approach. We find that the CLLP setting can be handled by an efficient algorithm, while some other forms of weak supervision may cause some parts of the problem to be *NP-hard*. We also show that the proposed new objective is closer to the true objective than a previous state-of-the-art method.

2. We propose a new proper approximation for the objective and propose an algorithm based on CCCP to solve the approximated problem.

3. We propose a 2-slack cutting plane algorithm to accelerate each iteration of CCCP, and give an upper bound on the number of iterations before termination.

2 Related Work

There are several related terminologies on different labeling settings of a learning task, including semi-supervised learning, multiple instance learning, and candidate label learning. Sometimes they are all generally called weakly supervised learning [18], as distinguished from traditional supervised learning requiring full and exact labels.

In semi-supervised learning (SSL) [43], a training set contains both labeled and unlabeled instances. Refs. [12] and [44] propose semi-supervised solutions for structured output learning, where some instances have exact and full labels while the remaining instances are unlabeled. Ref. [36] extends the method in [44] to incorporate domain knowledge as constraints, e.g., in POS tagging, a sentence should have at least one verb. Ref. [34] even allows a training instance itself to be partially labeled, e.g., some tokens in a sentence are labeled while the rest are unlabeled. The major difference between SSL and CLLP is: in SSL an output variable of an instance is either *exactly* labeled or unlabeled, while in CLLP the supervision is a set of candidate labels for each local part of the output variables, which dose not indicate the exact label but contains more information than when unlabeled.

Multiple instance learning (MIL) [6, 40] is a classical learning problem with weak supervision. In MIL, instances are grouped into bags, and labels are given at the bag level. The original MIL only admits a binary label for a bag, and is extended to admit multiple labels later [42, 16]. Some recent MIL methods consider the dependency among instances and bags, bringing the problem closer to structured output learning [41, 39, 5]. The difference between MIL and CLLP is: in MIL the label itself is accurate, but which instance deserves the label is ambiguous, while in CLLP the label itself is ambiguous (just a set of candidates) but which instance carries the label is clear.

Candidate label learning (CLL) [15, 11] assumes a set of candidate labels is given for each instance. It is later extended to the setting of *candidate labeling set* (CLS), where instances are grouped into bags, and for each bag a set of candidate labeling vectors is given [4, 14]. Each labeling vector consists of labels of all the instances in the bag. CLS looks similar with CLLP. However, CLS directly give candidate labeling vectors and has no constraints on the form of the candidate labeling set. This label setting may result in inefficiency, as shown in Theorem 1 of Section 3. We will discuss the relation between our approach and a state-of-the-art method of CLS [14] in Section 3.5, and make empirical comparisons under various tasks in Section 5.

We have noticed that in NLP literature, there are some papers on POS tagging with only POS dictionaries rather than concrete token-wise labels in sentences [23, 28, 8, 25, 10, 9], which is similar to the motivation of CLLP. However, they focus on the specific domain problem and may be difficult to extend to general structured prediction or multiclass classification. In contrast, in this paper we work on providing a general formulation and an efficient algorithm for this type of problems. The proposed approach is able to solve all kinds of structured predictions or multiclass classifications with the CLLP labeling setting.

3 Learning with Candidate Labels for Local Parts

3.1 General Weak Supervision via Candidate Labeling Set

Let $\mathbf{x} \in \mathcal{X}$ denote an instance and $\mathbf{y} \subseteq \mathcal{Y}$ denote the true label that is a structured object such as a sequence, a tree, etc. $\mathcal{Y}$ is the full label space for $\mathbf{x}$ without any constraints. $Y \subseteq \mathcal{Y}$ is weak supervision for $\mathbf{x}$. Generally Y can be represented as a set of all the allowed *full* labels for $\mathbf{x}$, which is named *candidate labeling set* (CLS) [4, 14]. We make the *agnostic* assumption that $\mathbf{y} \in Y$, then $\{\mathbf{y}\} \subseteq Y \subseteq \mathcal{Y}$. Given a set of weakly supervised training examples, $\{(\mathbf{x}_i, Y_i)\}_{i=1}^N$, the learning task is to learn a function $f : \mathbf{x} \mapsto \mathbf{y}$. Obviously, the task becomes supervised learning if $Y_i = \{\mathbf{y}_i\}$, $\forall i$, and degenerates to unsupervised learning when $Y_i = \mathcal{Y}_i$, $\forall i$.

Following the convention of structured output learning, we formulate function f by maximizing a mediate linear function $F(\mathbf{x}, \mathbf{y}; \mathbf{w})$ parameterized by $\mathbf{w}$, namely

$$f(\mathbf{x}; \mathbf{w}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}) \rangle, \tag{1}$$

where Ψ is a joint feature representation of inputs and outputs, which is flexibly designed to fit various applications.

For simplicity's sake, we use $\delta\Psi_i(\mathbf{y}, \mathbf{y}')$ to denote $\Psi(\mathbf{x}_i, \mathbf{y}) - \Psi(\mathbf{x}_i, \mathbf{y}')$. The value of $\langle \mathbf{w}, \delta\Psi_i(\mathbf{y}, \mathbf{y}') \rangle$ is the cost of predicting $\mathbf{y}$ instead of $\mathbf{y}'$ given input $\mathbf{x}_i$.

Although there could be various kinds of structures for $\mathbf{y}$ with different forms of $\Psi(\mathbf{x}, \mathbf{y})$, for the simplicity of the presentation, we focus on the special case where $\mathbf{y}$ forms a sequence. It is not hard to generalize this special structured case to other general structured and non-structured cases.

3.2 Candidate Labels for Local Parts (CLLP)

CLS is a general representation for weak supervision that has been used in previous methods [14, 4]. When dealing with structured output learning with the maximum margin approach, due to the huge number of constraints, the cutting plane method is usually employed to accelerate training [17]. In the cutting plane method, there should be an algorithm that is able to efficiently find the constraint that is most violated by the current solution. However, the following theorem shows that under the general CLS setting it is not possible to train efficiently:

Theorem 1. *Given a structured instance* $\mathbf{x}$ *and arbitrary candidate labeling set* Y*, there is no algorithm that can always find the most possible labels (in* Y *or not in* Y*) in poly(*$|\mathbf{x}|$*) time unless* $P = NP$*, where* $|\mathbf{x}|$ *is the length of* $\mathbf{x}$*.*

Proof. Please refer to the supplementary material for the proofs.

But if candidate labels are given only for local parts, there exists efficient algorithms that could find the most possible labels for a sequence among its candidate/non-candidate labeling sets, as stated in the following theorem:

Theorem 2. *If the candidate labels are given marginally by local parts, namely, each Y_i in $\{\mathbf{x}_i, Y_i\}_{i=1}^N$ has the form $Y_i = \{Y_{i1} \otimes Y_{i2} \otimes \ldots \otimes Y_{iM_i}\} \subseteq \mathcal{Y}$, where Y_{ij} is the set of candidate labels that $\mathbf{x}_{ij}$ could possibly take, among which only one is fully correct; $\mathbf{x}_{ij}$ is the j-th local part in $\mathbf{x}_i$ whose size is upper bounded by some constant; M_i is the number of local parts in $\mathbf{x}_i$, then in the sequence structured learning there is an efficient algorithm (modified Viterbi algorithm) that can find the most possible labels among candidate and non-candidate labeling sets.*

Please note that although this theorem is for the sequence structure, by extending the Viterbi algorithm to general Belief Propagation, it is straightforward to get the same conclusion for the general graph with a limited tree width.

3.3 Loss Function

We use a loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ to quantify the quality of a predictor, which has the following properties:

$$\Delta(\mathbf{y}, \mathbf{y}) = 0 \tag{2}$$

$$\Delta(\mathbf{y}, \mathbf{y}') > 0, \forall \mathbf{y} \neq \mathbf{y}' \tag{3}$$

$$\Delta(\mathbf{y}_1, \mathbf{y}_2) + \Delta(\mathbf{y}_2, \mathbf{y}_3) \geq \Delta(\mathbf{y}_1, \mathbf{y}_3), \forall \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3 \in \mathcal{Y}. \text{ (Triangle inequality)} \tag{4}$$

Among many loss functions $\Delta(\cdot, \cdot)$ satisfying the above properties, hamming loss and 0/1 loss are commonly used.

3.4 Large-Margin Formulation

The original structured SVM [32] is formulated as the following problem

$$\min_{\mathbf{w}} \sum_{i=1}^{N} C \left| \max_{\mathbf{y}_i' \in \mathcal{Y}} [\Delta(\mathbf{y}_i^*, \mathbf{y}_i') + \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}_i', \mathbf{y}_i^*) \rangle] \right|_+ + \frac{1}{2}\|\mathbf{w}\|^2. \tag{5}$$

where $|\cdot|_+$ denotes $\max\{\cdot, 0\}$ and $\mathbf{y}_i^*$ is the true label of $\mathbf{x}_i$. The formulation encourages a large margin between a true label and the runner up.

In CLLP, we are given candidate labels for each local part, which has two implications: (1) any label in the non-candidate set is not the true label; (2) some label in the candidate set is true label but we do not know which one. They imply two different types of discriminative constraints that need consideration. First, discrimination between the candidates and non-candidates. Second, discrimination between the true label and other candidates. Thus we decompose the slacks for each instance into two sets, one set for candidate labels and another for non-candidate labels. Namely, we decompose the objective as

$$\begin{aligned} \mathcal{J}_0(\mathbf{w}) = & \sum_{i=1}^{N} C_1 \left| \max_{\mathbf{y}_i' \in Y_i} [\Delta(\mathbf{y}_i^*, \mathbf{y}_i') + \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}_i', \mathbf{y}_i^*) \rangle] \right|_+ \\ & + \sum_{i=1}^{N} C_2 \left| \max_{\mathbf{y}_i'' \in \mathcal{Y}/Y_i} [\Delta(\mathbf{y}_i^*, \mathbf{y}_i'') + \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}_i'', \mathbf{y}_i^*) \rangle] \right|_+ + \frac{1}{2}\|\mathbf{w}\|^2. \end{aligned} \tag{6}$$

However, in contrast to the supervised case, in CLLP the true labels $\mathbf{y}_i^*$'s are unknown. We can estimate them to approximate the objective. Thus our optimization problem becomes

$$\min_{\mathbf{w},\{\mathbf{y}_i\in Y_i\}_{i=1}^N} \mathcal{J}_c(\mathbf{w},\{\mathbf{y}_i\}_{i=1}^N) = \sum_{i=1}^N C_1 \left| \max_{\mathbf{y}_i'\in Y_i} [\Delta(\mathbf{y}_i,\mathbf{y}_i') + \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}_i',\mathbf{y}_i)\rangle] \right|_+ + \sum_{i=1}^N C_2 \left| \max_{\mathbf{y}_i''\in \mathcal{Y}/Y_i} [\Delta(\mathbf{y}_i,\mathbf{y}_i'') + \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}_i'',\mathbf{y}_i)\rangle] \right|_+ + \frac{1}{2}\|\mathbf{w}\|^2, \quad (7)$$

where $\mathbf{y}_i$ is the estimation of the true label $\mathbf{y}_i^*$. The intuition is that we encourage a large margin between the estimated "true" labels and the runner up in the candidate labeling set as well as another runner up in the non-candidate set. And we differentiate these two margins by C_1 and C_2.

Equation (7) looks similar to the counterparts in the Transductive Struct-SVMs [44] and the Latent Struct-SVMs [37]. However, there are three key differences. First, we do not know any true label $\mathbf{y}_i^*$ in Equation (7), while in the Transductive Struct-SVMs we know the true labels of the labeled set and in the Latent Struct-SVMs we know the true labels of the observed layer. Second, we differentiate the two types of large margin constraints. Third, in our problem, each $\mathbf{y}_i$ has its own feasible solution space Y_i.

3.5 Properties of the Objective

We compare our objective with the true objective and another objective used in the current state-of-the-art method, the Maximum Margin Set learning (MMS) [14] designed for the CLS setting.

Lemma 1. *$\forall \mathbf{w}$, $\mathcal{J}_0(\mathbf{w}) \geq \min_{\{\mathbf{y}_i\in Y_i\}_{i=1}^N} \mathcal{J}_c(\mathbf{w},\{\mathbf{y}_i\}_{i=1}^N)$. Namely, the objective Equation (6) upper bounds the objective Equation (7).*

Corollary 1. *Let $\mathcal{J}_0^* = \min_{\mathbf{w}} \mathcal{J}_0(\mathbf{w})$, and $\mathcal{J}_c^* = \min_{\mathbf{w},\{\mathbf{y}_i\in Y_i\}_{i=1}^N} \mathcal{J}_c(\mathbf{w},\{\mathbf{y}_i\}_{i=1}^N)$, then $\mathcal{J}_0^* \geq \mathcal{J}_c^*$. Namely, the optimal value of the objective Equation (6) upper bounds that of the objective Equation (7).*

On the other hand, in [14], the MMS method is proposed for tackling multi-class classification with candidate labeling sets. Actually, it can be straightforwardly extended to a structured output case by modifying $\Delta(\cdot,\cdot)$ and $\Psi(\cdot,\cdot)$. Then the problem of MMS becomes:

$$\min_{\mathbf{w}} \mathcal{J}_m(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C_2 \sum_{i=1}^N \left| \max_{\mathbf{y}_i''\notin Y_i} [\Delta_{\min}(\mathbf{y}_i'', \mathcal{Y}/Y_i) + \langle \mathbf{w}, \Psi(\mathbf{x}_i,\mathbf{y}_i'')\rangle] - \max_{\mathbf{y}_i\in Y_i} \langle \mathbf{w}, \Psi(\mathbf{x}_i,\mathbf{y}_i)\rangle \right|_+ \quad (8)$$

where $\Delta_{\min}(\mathbf{y}', Y) = \min_{\mathbf{y}\in Y} \Delta(\mathbf{y}',\mathbf{y})$. Then we have the following lemma:

Lemma 2. *$\forall \mathbf{w}, \min_{\{\mathbf{y}_i\in Y_i\}_{i=1}^N} \mathcal{J}_c(\mathbf{w},\{\mathbf{y}_i\}_{i=1}^N) \geq \mathcal{J}_m(\mathbf{w})$. Namely, the objective Equation (7) upper bounds the objective Equation (8).*

Corollary 2. *Let* $\mathcal{J}_c^* = \min_{\mathbf{w},\{\mathbf{y}_i \in Y_i\}_{i=1}^N} \mathcal{J}_c(\mathbf{w}, \{\mathbf{y}_i\}_{i=1}^N)$*, and* $\mathcal{J}_m^* = \min_{\mathbf{w}} \mathcal{J}_m(\mathbf{w})$*, then* $\mathcal{J}_c^* \geq \mathcal{J}_m^*$*. Namely, the optimal value of the objective Equation (7) upper bounds that of the objective Equation (8).*

By combining the above lemmas and corollaries, we obtain the theorem:

Theorem 3. $\forall \mathbf{w}, \mathcal{J}_0(\mathbf{w}) \geq \min_{\{\mathbf{y}_i \in Y_i\}_{i=1}^N} \mathcal{J}_c(\mathbf{w}, \{\mathbf{y}_i\}_{i=1}^N) \geq \mathcal{J}_m(\mathbf{w})$ *and*

$$\mathcal{J}_0^* \geq \mathcal{J}_c^* \geq \mathcal{J}_m^*.$$

This theorem shows that the value of our objective (Equation (7)) lies between the value of the true objective (Equation (6)) and the value of the objective given by MMS (Equation (8)), indicating that our objective is closer to the true objective compared to MMS.

4 Optimization

4.1 Optimization with CCCP

The optimization problem defined by Equation (7) is non-convex. An effective approach to solving such a non-convex problem is the Constrained Concave-Convex Procedure (CCCP) [38, 29], which requires the objective to be decomposed into a convex part and a concave part. However, the objective of Equation (7) is hard to decompose. In Equation (7), we maximize the objective with $\mathbf{y}_i'$ while minimizing it with $\mathbf{y}_i$. But the term $\Delta(\mathbf{y}_i, \mathbf{y}_i')$ correlates them together, obstructing the decomposition. The same problem exists with the term $\Delta(\mathbf{y}_i, \mathbf{y}_i'')$. Therefore, we make an approximation of the objective by decomposing each $\Delta(a, b)$ term into $(\Delta(a, c) + \Delta(c, b))$, resulting in the following objective:

$$\min_{\mathbf{w}} \sum_i \min_{\mathbf{y}_i \in Y_i} \left\{ C_1 \left| \max_{\mathbf{y}_i' \in Y_i} [\Delta(\mathbf{z}_i, \mathbf{y}_i') + \Delta(\mathbf{z}_i, \mathbf{y}_i) + \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}_i', \mathbf{y}_i)\rangle] \right|_+ + \right.$$
$$\left. C_2 \left| \max_{\mathbf{y}_i'' \in \mathcal{Y}/Y_i} [\Delta(\mathbf{z}_i, \mathbf{y}_i'') + \Delta(\mathbf{z}_i, \mathbf{y}_i) + \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}_i'', \mathbf{y}_i)\rangle] \right|_+ \right\} + \frac{1}{2}\|\mathbf{w}\|^2 \tag{9}$$

$$= \min_{\mathbf{w}} \sum_i \left\{ C_1 \left| \max_{\mathbf{y}_i' \in Y_i} [\Delta(\mathbf{z}_i, \mathbf{y}_i') + \langle \mathbf{w}, \Psi_i(\mathbf{x}_i, \mathbf{y}_i')\rangle] - \max_{\mathbf{y}_i \in Y_i} [\langle \mathbf{w}, \Psi_i(\mathbf{x}_i, \mathbf{y}_i)\rangle - \Delta(\mathbf{z}_i, \mathbf{y}_i)] \right|_+ + \right.$$
$$\left. C_2 \left| \max_{\mathbf{y}_i'' \in \mathcal{Y}/Y_i} [\Delta(\mathbf{z}_i, \mathbf{y}_i'') + \langle \mathbf{w}, \Psi_i(\mathbf{x}_i, \mathbf{y}_i'')\rangle] - \max_{\mathbf{y}_i \in Y_i} [\langle \mathbf{w}, \Psi_i(\mathbf{x}_i, \mathbf{y}_i)\rangle - \Delta(\mathbf{z}_i, \mathbf{y}_i)] \right|_+ \right\} + \frac{1}{2}\|\mathbf{w}\|^2 \tag{10}$$

where $\mathbf{z}_i$'s are labels initialized at the beginning of each CCCP iteration. As $\Delta(\cdot, \cdot)$ meets the triangle inequality, Equation (10) upper bounds Equation (7).

Now we can construct an upper bound for the concave term. In each CCCP iteration we substitute the concave term

$$\max_{\mathbf{y}_i \in Y_i} [\langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}_i)\rangle - \Delta(\mathbf{z}_i, \mathbf{y}_i)] \tag{11}$$

with term $\langle \mathbf{w}, \Psi(\mathbf{x}_i, \overline{\mathbf{y}_i})\rangle - \Delta(\mathbf{z}_i, \overline{\mathbf{y}_i})$, where

Algorithm 1 The CCCP algorithm

1: **Input:** data with weak supervision $\{(\mathbf{x}_i, Y_i)\}_{i=1}^N\}$
2: Initialize labels $\{\overline{\mathbf{y}_i}\}_{i=1}^N$
3: **repeat**
4: Solve the convex optimization problem given by Equation (15)
5: Set labels $\{\overline{\mathbf{y}_i}\}_{i=1}^N$ to be the current prediction of structured instances $\{\mathbf{x}_i\}_{i=1}^N$ given by current model
6: **until** convergence to a local minimum

$$\overline{\mathbf{y}_i} = \arg\max_{\mathbf{y}_i \in Y_i} [\langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}_i)\rangle - \Delta(\mathbf{z}_i, \mathbf{y}_i)] \tag{12}$$

At the beginning of each CCCP iteration we initialize

$$\mathbf{z}_i = \arg\max_{\mathbf{z}_i \in Y_i} \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{z}_i)\rangle \tag{13}$$

Then it follows that $\overline{\mathbf{y}_i} = \mathbf{z}_i$, indicating that we could directly initialize $\overline{\mathbf{y}_i}$'s as

$$\overline{\mathbf{y}_i} = \arg\max_{\mathbf{y}_i \in Y_i} \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}_i)\rangle. \tag{14}$$

In this way, we are essentially setting $\overline{\mathbf{y}_i}$ to be the predicted labels of structured instances given by current model. Then the optimization problem in each iteration of CCCP becomes

$$\begin{aligned} \min_{\mathbf{w}} \sum_{i=1}^{N} \Bigg[& -(C_1 + C_2)\langle \mathbf{w}, \Psi(\mathbf{x}_i, \overline{\mathbf{y}_i})\rangle + C_1 \cdot \max_{\mathbf{y}_i' \in Y_i} (\Delta(\overline{\mathbf{y}_i}, \mathbf{y}_i') + \langle \mathbf{w}, \Psi(\mathbf{x}_i, \mathbf{y}_i')\rangle) \\ & + C_2 \cdot \max_{\mathbf{y}_i'' \in \mathcal{Y}/Y_i \cup \{\overline{\mathbf{y}_i}\}} (\Delta(\overline{\mathbf{y}_i}, \mathbf{y}_i'') + \langle \mathbf{w}, \Psi(\mathbf{x}_i \mathbf{y}_i'')\rangle) \Bigg] \end{aligned} \tag{15}$$

where $\overline{\mathbf{y}_i}$'s are initialized as Equation (14). The CCCP procedure is shown in Algorithm 1.

4.2 Accelerating with 2-Slack Cutting Plane Algorithm

In each iteration the optimization problem of Equation (15) can be solved using standard quadratic programming. However, the huge number of constraints prevents us from solving it efficiently. We employ the Cutting Plane (CP) algorithm [19, 17] to accelerate training. However, in contrast to the original CP for structured SVM [17], in this problem we have two sets of constraints (corresponding to C_1 and C_2 respectively) and we want to find the solution that satisfies them with specified precision separately. Thus we need to maintain two constraint sets Ω_1 and Ω_2, and set two precision ε_1 and ε_2 for them respectively. To find the most violated label setting in candidate labeling sets and non-candidate labeling sets, we employ a modified *Viterbi* algorithm, which will run in polynomial time of $|\mathbf{x}|$ for each instance $\mathbf{x}$ (For details of the modified Viterbi algorithm please refer to the supplementary material). The sketch of the 2-slack cutting plane algorithm is described in Algorithm 1 in the supplementary material. We also show that the algorithm will converge in at most a non-trivial fixed number of iterations. For the details please refer to Theorems 4 & 5 in Section 4 of the supplementary material.

5 Experiments

We performed experiments on three sequence labeling tasks including part-of-speech tagging (POS), chunking (CHK) and bio-entity recognition (BNE).

5.1 Tasks & Data Sets

POS: This task aims to assign each word of a sentence a unique tag indicating its linguistic category such as noun, verb, etc. We used the *Penn Treebank* [22] corpus with the parts extracted from the *Wall Street Joural (WSJ)* in 1989.
CHK: This task aims to divide a sentence into constituents that are syntactic groups such as noun groups, verb groups etc. We use the same data set in the shared task of Chunking in CoNLL 2000 [3] [30].
BNE: This task aims to identify technical terms and tag them in some predefined categories. We used the same dataset in the Bio-Entity Recognition Task at BioNLP/NLPBA 2004 [4] [20].

5.2 Baseline Methods

Our method, denoted by CLLP, was implemented based on the SVM$^{\text{hmm}}$ package[5] to fit with sequence labeling tasks. We compared CLLP with the following methods that are able to handle sequences with candidate labels:
Gold: We trained an SVM$^{\text{hmm}}$ predictor with ground truth full labels. The performance of Gold would be an upper bound of the performance of CLLP.
NAIVE: For each token, we randomly picked one label from its candidate labels as its true label and trained a SVM$^{\text{hmm}}$ predictor.
CL-SVM$^{\text{hmm}}$: We treated all the candidate labels as true labels. Each sequence appeared in the training set multiple times with different labels. Then an SVM$^{\text{hmm}}$ predictor was trained on the self-contradictory labeled sequences. Similar methods have been used as baselines in [2, 14].
MMS: This method was originally proposed in [14]. We made modifications as stated in Section 4 to deal with the sequence data.

All of the above methods were implemented based on the SVM$^{\text{hmm}}$ package. For all the experiments, we selected cost parameters C, C_1 and C_2 from the grids $[500 : 150 : 3200]$. In MMS and CLLP, each CCCP iteration was a cutting plane optimization procedure whose iteration number was controlled by the parameters ε (for MMS) and ε_1 and ε_2 (for CLLP). Training too aggressively (with ε's that are too small) in the first several CCCP iterations would prevent the algorithm from recovering from the wrongly initialized labels. Thus we initialized ε (for MMS) and ε_1 and ε_2 (for CLLP) to be large at first, and then divided ε's by a discounter d in each iteration until they were less than or equal to some thresholds t, t_1 and t_2. We set the discounter to be 2 and choose thresholds from grids $[0.5 : 0.5 : 3]$.

[3] http://www.clips.ua.ac.be/conll2000/chunking/
[4] http://www.nactem.ac.uk/tsujii/GENIA/ERtask/report.html
[5] http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html

5.3 Experiments on Artificial Candidate Labels

Originally, these three data sets did not contain any candidate labels as supervision. We followed [14] to generate artificial candidate labels for them. In this way we were able to perform controlled experiments and study the impact of different labeling settings such as the size of the candidate set.

Candidate Label Generation

The following two methods were adopted to generate candidate labels. For both,we took each individual token as a local part, i.e., we provided candidate labels for each token, where the number of candidate labels was specified by the token's candidate labeling size. The two methods were used to control label ambiguity at the sequence level and token level respectively.

Random Generation: This method was used to control the label ambiguity at the sequence level. Each token in the sequence had an initial candidate labeling size of 1 (which is its true label). We randomly chose n tokens sequentially (not necessarily non-overlapping) and doubled their candidate labeling size. We then generated candidate labels for each token according to the label distribution in the training data, which already contained label bias.

Specified Generation: This method was used to control the label ambiguity at the token level. For all sequences, we restricted all the candidate labeling sizes to be a constant m, and randomly generated m *different* candidate labels for each token, among which only one was correct.

The NAIVE, MMS and CLLP methods require label initializations. We randomly picked one label from the candidate labels for each token as its initial label.

Results

Data Sets with Random Generation

We varied n from 1 to 16. Performances of various methods on 3 different data sets were plotted in Figure 1, from which we can make observations:

First, CLLP was more stable against different numbers of candidate labels compared to NAIVE and CL-SVM$^{\text{hmm}}$. In addition, CL-SVM$^{\text{hmm}}$ was not scalable with a large number of candidate labels. When n exceeds 6, several days are needed for training.

Second, the gap between CLLP and MMS was small, especially with regards to CHK. This phenomenon resulted from the small number of candidate labels. With the random generation of candidate labels, even when n was large, there were still tokens that had only one candidate label that was exactly its true label. This fact prevented CLLP from taking advantage of its objective of better approximation then MMS, and made the gap between them negligible. However, the gap will be more visible when the number of candidate labels is large, as shall be seen in Figure 2 of Section 5.3.

Last, the CLLP method beats all the other methods and performed close to the full supervised SVM$^{\text{hmm}}$. This clearly shows the effectiveness and scalability of CLLP versus other methods.

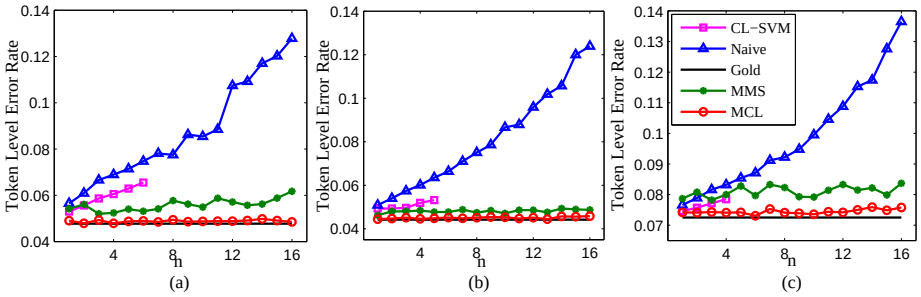

Fig. 1. The performances of various methods on data sets POS (a), CHK (b) and BNE (c). We only plotted a few points of CL-SVM$^{\text{hmm}}$ because as the number of candidate label grows, it immediately becomes unfeasible in time.

Data Sets with Specified Generation

We varied m from 1 to 7 to see how the token-wise candidate labeling size affected the performance. We report the results on the CHK data set in Figure 2 (a).

The results indicate the gap between MMS and CLLP becomes more visible as m increases. This phenomenon mainly results from poor approximation of the objective of MMS.

In the objective 8, MMS considered only the margin between the most possible candidate labels and the most possible non-candidate labels. When the number of candidate labels was large, there were fewer non-candidate labels for the MMS optimizer to choose constraints from. In contrast, CLLP considers constraints from both the candidate labeling set and the non-candidate labeling set. This strategy is beneficial when the number of candidate labels is large.

We also observed that CLLP was less sensitive to the initialization error when compared to other methods, Due to the fact that initialization error increases as m increases. We conducted an auxiliary experiment at $m = 5$ to further investigate. We varied the initialization error rate from 0.5 to 0.8. Note that when the initialization error was 0.8, the initial labels were actually totally random. Under this setting, the performances of various methods are reported in Figure 2 (b). Based on the results, the initialization error rate did have a significant impact on the performances of these models. However, its influence on CLLP was limited compared to other methods, showing the stability of CLLP against the different initialization error rates.

Impact of Parameter d

We conducted this experiment to verify the impact of the discounter d on the convergence of CLLP. The experiment was done on the POS data set with random generation and n was fixed at 16. We varied the number of iterations from 1 to 10, and set the discounter d to grids $[2 : 5]$ to see the impact.The results are reported in Figure 3.

The results show the algorithm converged quickly, e.g., in 4 to 5 iterations. Even if we chose an inappropriate d (say, $d = 2$), the algorithm still converged in

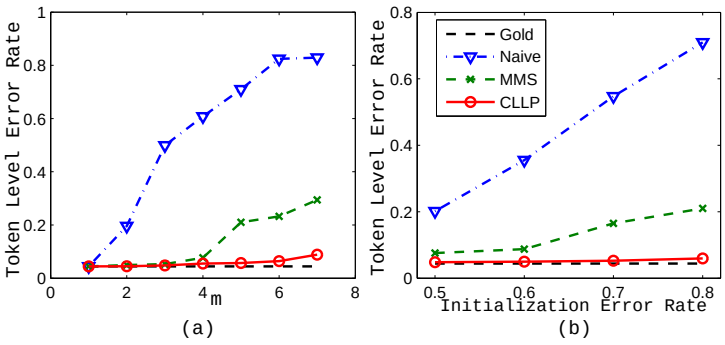

Fig. 2. Impact of model parameters on the performances of various methods. (a) Impact of m with totally random label initialization. (b) Impact of initialization error rate with $m = 5$.

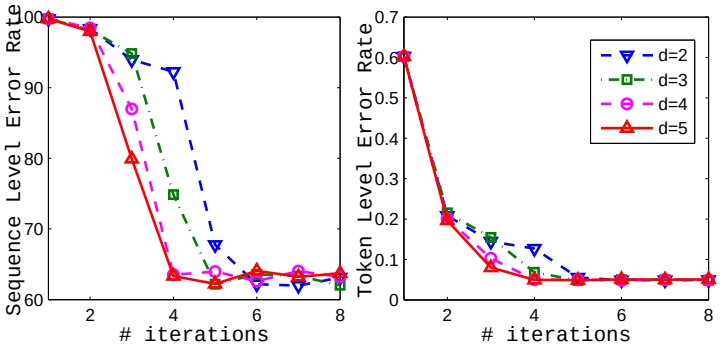

Fig. 3. The convergence curves of CLLP with different discounters d.

6 iterations, showing the efficiency and robustness of the algorithm. The speed of convergence seems to be positively correlated with the value of d. However, the impact was limited. Actually it made little sense to choose a very large value for d, since the algorithm would simply do nothing in the first several CCCP iterations and would set wrong labels for the following training procedure.

5.4 Real Application: POS Tagging with Dictionaries

We also conducted an experiment on the real application of POS tagging with dictionaries. Our goal was to train a POS tagger without any labeled sentences but only require a word dictionary indicating the possible POS tags of each word, which is easy to obtain from various kinds of linguistic dictionaries. This problem has been studied before in NLP and some specific methods have been proposed [23, 28, 25, 9]. We noticed that this is a typical example of structured output learning with CLLP: by matching the dictionary back to each sentence, we obtained the candidate POS tags for the matched words. We found that our

Table 1. POS tagging accuracy of different methods.

Methods	48000	96000
GOLD	94.40	94.77
NAIVE	65.33	66.79
MMS	67.68	69.51
CLLP	**76.74**	**76.56**
MinGreedy [25]	68.86	74.93
MinGreedy + auto-supervised [9]	80.78	80.90
MinGreedy + auto-supervised + emission initialization [9]	80.92	80.70
MinGreedy with extensions [9]	87.95	86.22
CLLP + auto-supervised	82.90	82.22
CLLP + auto-supervised + emission initialization	82.89	82.22
CLLP + MinGreedy with extensions	**89.87**	**88.47**

general algorithm is competitive with the state-of-the-art methods, i.e., "MinGreedy" [25] and its various extensions [9].

Following the settings in previous methods, we used a corpus extracted from the *Wall Street Joural (WSJ)* in 1989 in *Penn Treebank* [22]. Sections 00-07, with golden labels, were used to construct the tag dictionary. Then the first 48000/96000 words of sections 16-18 without any labels were used as the raw training set. Sections 19-21 were used as the development set and 22-24 as the test set.

In standard CLLP, the initial labels are randomly chosen from the candidate labels, without any task-specific prior incorporated into the algorithm. In [9], several ways of label initializations have been proposed. We used some of these methods to initialize labels for CLLP, noted as "CLLP + auto-supervised" and "CLLP + auto-supervised + emission initialization." The best performance shown in [9] was achieved by MinGreedy with full extensions (method 10 in the original paper), where a full-supervised HMM was trained using initialized labels output by MinGreedy. We also used the output of MinGreedy as our initialization for CLLP, noted as "CLLP + MinGreedy with extensions." The MinGreedy code is provided by the authors [6]. More details on dictionary construction and label initialization can be found in [25, 9].

The results are shown in Tabel 1. CLLP outperformed all the other unitary methods without the task (POS) specific initializations. With the proper initialization of labels, CLLP is able to further improve the results. Remarkably, by using the output labels of MinGreedy with full extensions as label initialization, CLLP is able to outperform all the other methods.

5.5 Real Application: Wiki Entity Type Disambiguation

We conducted another experiment on a real problem of Wiki entity type disambiguation. This is an example of CLLP degenerating to handle multiclass classification.

[6] `https://github.com/dhgarrette/type-supervised-tagging-2012emnlp`

Traditionally, in order to train an NER model, we need sentence-level labels. Similar to POS tagging with dictionaries, we attempted to train an NER model without any sentence labels and only requiring an entity dictionary indicating the possible types of an entity, which can be easily obtained from many knowledge bases such as Freebase [7].

We conducted the experiments based on Freebase and Wikipedia articles. In some sentences of a Wikipedia article, there is an anchor link to indicate the phrase is an entity in Wikipedia and will redirect to the host article page if a user clicks it. For each entity highlighted by the anchor link, we can find the corresponding entity types in Freebase. We then obtained a set of multiclass training instances: an entity in a sentence, and the corresponding candidate labels. We selected 20 entity types plus an "Other" type. We randomly sampled 500 entities to manually label as the test set, and sampled 9991 entities as the training set. As each entity was associated with a candidate "Other" label, the "Other" class dominated other classes. Thus we sampled the "Other" class by assigning "Other" to an entity with a probability of 0.1. The the classes were therefore more balanced.

Table 2. Results on Wiki data

	F1	Precision	Recall
NAIVE	64.83	57.79	73.82
MMS	54.02	47.70	62.30
CLLP	**69.69**	**61.52**	**80.37**

The results are shown in Tabel 2. We found that MMS was even worse than NAIVE. The main reason for this phenomenon still draws from the objective that MMS aims to minimize. With the large number of "Others" labels, MMS either took it as the most possible candidate label, or simply ignored it since it was not a non-candidate label. Thus it proned to predicting many entities to "Others." Things became even worse when we added more "Others" labels to the training data. When we associated all the instances with an extra "Others" label, MMS simply predicted all the entities were "Others." In contrast, CLLP overcame this problem by using two sets of constraints and outperformed the other two methods.

6 Conclusion

In this paper, we introduced a new weakly supervised setting for structured output learning, named *candidate labels for local parts* (CLLP), where a set of candidate labels is provided for each local part of output variables. We have shown that training with this type of weak supervision can be efficiently handled. Then we proposed a large-margin formulation for the learning problem, and used proper approximations and Constraint Concave-Convex Procedure (CCCP) to

[7] `http://www.freebase.com`

deal with the non-convex optimization problem. A 2-slack cutting plane method has also been proposed to accelerate the inner loop of CCCP. Experiments on various tasks have shown the effectiveness and efficiency of the proposed method. It is interesting that the CLLP setting is rather general, and is able to degenerate to various weakly supervised setting for both structured output learning and multiclass classification. Thus the CLLP setting and the proposed large-margin learning method provide a uniform approach to formulate and solve structured output learning with different kinds of weak supervision.

References

1. Barnard, K., Duygulu, P., Forsyth, D., De Freitas, N., Blei, D., Jordan, M.: Matching words and pictures. JMLR pp. 1107–1135 (2003)
2. Bunescu, R., Mooney, R.: Multiple instance learning for sparse positive bags. In: ICML. pp. 105–112 (2007)
3. Chang, M., Goldwasser, D., Roth, D., Srikumar, V.: Structured output learning with indirect supervision. In: ICML (2010)
4. Cour, T., Sapp, B., Jordan, C., Taskar, B.: Learning from ambiguously labeled images. In: CVPR. pp. 919–926 (2009)
5. Deselaers, T., Ferrari, V.: A conditional random field for multiple-instance learning. In: ICML. pp. 287–294 (2010)
6. Dietterich, T., Lathrop, R., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence pp. 31–71 (1997)
7. Dredze, M., Talukdar, P., Crammer, K.: Sequence learning from data with multiple labels. In: ECML/PKDD Workshop on Learning from Multi-Label Data (2009)
8. Ganchev, K., Gillenwater, J., Taskar, B.: Dependency grammar induction via bitext projection constraints. In: ACL. pp. 369–377 (2009)
9. Garrette, D., Baldridge, J.: Type-supervised hidden markov models for part-of-speech tagging with incomplete tag dictionaries. In: EMNLP. pp. 821–831 (2012)
10. Hall, K., McDonald, R., Katz-Brown, J., Ringgaard, M.: Training dependency parsers by jointly optimizing multiple objectives. In: EMNLP. pp. 1489–1499 (2011)
11. Hüllermeier, E., Beringer, J.: Learning from ambiguously labeled examples. Intelligent Data Analysis pp. 419–439 (2006)
12. Jiao, F., Wang, S., Lee, C.H., Greiner, R., Schuurmans, D.: Semi-supervised conditional random fields for improved sequence segmentation and labeling. In: ACL. pp. 209–216 (2006)
13. Jie, L., Caputo, B., Ferrari, V.: Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In: NIPS, pp. 1168–1176 (2009)
14. Jie, L., Francesco, O.: Learning from candidate labeling sets. In: NIPS, pp. 1504–1512 (2010)
15. Jin, R., Ghahramani, Z.: Learning with multiple labels. NIPS pp. 897–904 (2002)
16. Jin, R., Wang, S., Zhou, Z.: Learning a distance metric from multi-instance multi-label data. In: CVPR. pp. 896–902 (2009)
17. Joachims, T., Finley, T., Yu, C.: Cutting-plane training of structural svms. Machine Learning pp. 27–59 (2009)
18. Joulin, A., Bach, F.: A convex relaxation for weakly supervised classifiers. In: ICML (2012)
19. Kelley Jr, J.: The cutting-plane method for solving convex programs. Journal of the Society for Industrial & Applied Mathematics pp. 703–712 (1960)

20. Kim, J.D., Ohta, T., Tsuruoka, Y., Tateisi, Y., Collier, N.: Introduction to the bio-entity recognition task at jnlpba. In: JNLPBA. pp. 70–75 (2004)
21. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML. pp. 282–289 (2001)
22. Marcus, M., Santorini, B., Marcinkiewicz, M.: Building a large annotated corpus of english: The penn treebank. Computational Linguistics pp. 313–330 (1993)
23. Merialdo, B.: Tagging english text with a probabilistic model. Computational linguistics pp. 155–171 (1994)
24. Navigli, R.: Word sense disambiguation: A survey. ACM Comput. Surv. 41(2), 10:1–10:69 (Feb 2009)
25. Ravi, S., Vaswani, A., Knight, K., Chiang, D.: Fast, greedy model minimization for unsupervised tagging. In: COLING. pp. 940–948 (2010)
26. Roth, D., Small, K.: Margin-based active learning for structured output spaces. ECML pp. 413–424 (2006)
27. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: EMNLP. pp. 1070–1079 (2008)
28. Smith, N.A., Eisner, J.: Contrastive estimation: Training log-linear models on unlabeled data. In: ACL. pp. 354–362 (2005)
29. Smola, A.J., Vishwanathan, S., Hofmann, T.: Kernel methods for missing variables. In: AISTATS (2005)
30. Tjong K. S., E., Buchholz, S.: Introduction to the conll-2000 shared task: Chunking. In: CoNLL. pp. 127–132 (2000)
31. Tjong K. S., E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: HLT-NAACL. pp. 142–147 (2003)
32. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML (2004)
33. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. JMLR pp. 1453–1484 (2005)
34. Tsuboi, Y., Kashima, H., Oda, H., Mori, S., Matsumoto, Y.: Training conditional random fields using incomplete annotations. In: COLING. pp. 897–904 (2008)
35. Vezhnevets, A., Ferrari, V., Buhmann, J.: Weakly supervised structured output learning for semantic segmentation. In: CVPR. pp. 845–852 (2012)
36. Yu, C.N.: Transductive learning of structural svms via prior knowledge constraints. In: AISTATS (2012)
37. Yu, C.N.J., Joachims, T.: Learning structural svms with latent variables. In: ICML. pp. 1169–1176 (2009)
38. Yuille, A., Rangarajan, A.: The concave-convex procedure. Neural Computation pp. 915–936 (2003)
39. Zhang, D., Liu, Y., Si, L., Zhang, J., Lawrence, R.: Multiple instance learning on structured data. In: NIPS, pp. 145–153 (2011)
40. Zhou, Z.: Multi-instance learning: A survey. Tech. rep., AI Lab, Department of Computer Science & Technology, Nanjing University (Mar 2004)
41. Zhou, Z., Sun, Y., Li, Y.: Multi-instance learning by treating instances as non-iid samples. In: ICML. pp. 1249–1256 (2009)
42. Zhou, Z., Zhang, M.: Multi-instance multi-label learning with application to scene classification. In: NIPS. pp. 1609–1616 (2006)
43. Zhu, X.: Semi-supervised learning literature survey (2005)
44. Zien, A., Brefeld, U., Scheffer, T.: Transductive support vector machines for structured variables. In: ICML (2007)